SII

DPU-S445 シリーズ
Microsoft® .NET™ Compact Framework
プリントクラスライブラリ
技術説明書

セイコーインスツル株式会社

2008 年　9 月

© セイコーインスツル株式会社　2008
無断転載を禁じます。
本書の内容は、断りなく変更することがあります。
この技術説明書を第三者に無断で頒布することを禁じます。
本書および本書に記載された製品の利用によって発生した損害
およびその回復に要する費用に対し、当社は一切の責任を負いかねます。
Microsoft®、Windows®、Visual Studio®、Visual Basic®、Visual C#®、.NET™は、
米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。

はじめに

本書は、セイコーインスツル（以下、SII）製サーマルプリンタ DPU-S445 シリーズ用に提供する Microsoft® .NET Compact Framework プリントクラスライブラリ SII DPU-S445 NETCF10、および、SII DPU-S445 NETCF20 の仕様、機能に関する情報について説明します。

目次

はじめに　i

1章　製品概要　1-1

1.1　本ライブラリの提供機能　1-1
1.2　対応する.NET Compact Framework　1-2
1.2.1　.NET Compact Framework のバージョンと対応製品　1-2

2章　製品仕様　2-1

2.1　製品仕様　2-1
2.1.1　動作環境　2-1
2.1.2　通信方式の詳細　2-2
2.2　提供ファイルと使用方法　2-3
2.2.1　提供ファイル　2-3
2.2.2　本ライブラリのプロジェクトへの組み込み　2-3
2.2.3　作成したアプリケーションのターゲットデバイスでの実行方法　2-3
2.2.4　その他　2-3

3章　インストール　3-1

3.1　インストーラ　3-1
3.1.1　インストーラについて　3-1
3.1.2　インストールファイルについて　3-2

4章　プリントクラスライブラリの機能　4-1

4.1　クラスの概要　4-1
4.1.1　ネームスペースとクラス名　4-1
4.1.2　クラスの構造　4-2
4.1.3　メソッド一覧　4-3
4.1.4　プロパティ一覧　4-3
4.2　メソッド詳細　4-4
4.2.1　コンストラクタ　4-4
4.2.2　Connect メソッド　4-5
4.2.3　Disconnect メソッド　4-7

4.2.4　GetStatus メソッド　4-8
4.2.5　SendText メソッド　4-10
4.2.6　SendBinary メソッド　4-11
4.2.7　SendDataFile メソッド　4-12
4.2.8　Abort メソッド　4-15
4.3　プロパティ詳細　4-16
4.3.1　SendTimeout プロパティ　4-16
4.3.2　ReceiveTimeout プロパティ　4-17
4.3.3　CountryCode プロパティ　4-18
4.3.4　Baudrate プロパティ　4-20
4.3.5　Handshake プロパティ　4-21
4.3.6　ByteSize プロパティ　4-22
4.3.7　StopBits プロパティ　4-23
4.3.8　Parity プロパティ　4-24
4.3.9　PortName プロパティ　4-25
4.3.10　PortType プロパティ　4-26
4.3.11　IsOpen プロパティ　4-27

5章　サンプルプログラム　5-1

5.1　提供サンプル　5-1
5.1.1　サンプルプログラム　5-1
5.1.2　サンプルプログラムのインストール　5-2
5.1.3　サンプルプログラムの実行　5-3
5.1.4　サンプルプログラムのソースファイル　5-5
5.1.5　注意事項　5-5

6章　免責　6-1

1章　製品概要

本章では、SII DPU-S445 NETCF10、および、SII DPU-S445 NETCF20（以下、本ライブラリ）の製品概要について説明します。

1.1　本ライブラリの提供機能

本ライブラリは、Microsoft® .NET Compact Framework（以下、.NET Compact Framework）環境で動作するアプリケーションに SII 製プリンタ DPU-S445 シリーズ（以下、プリンタ）を利用するための機能を提供します。（図 1-1）

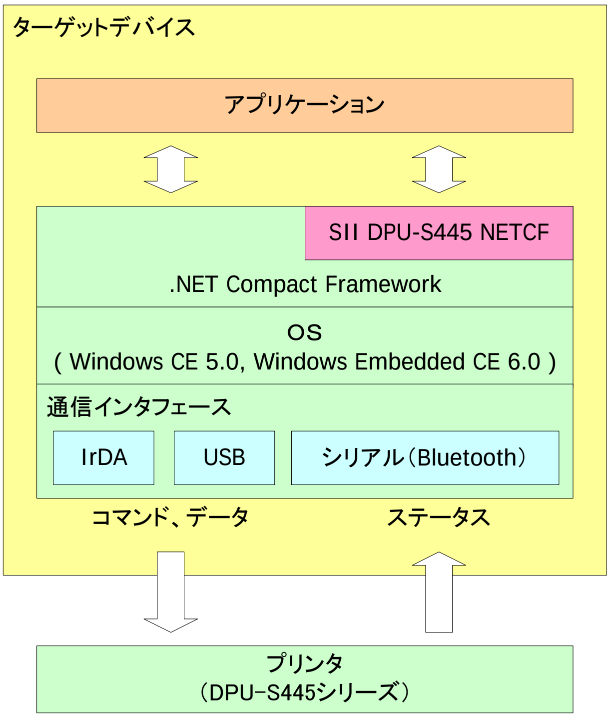


図 1-1

アプリケーションは、本ライブラリにより、ターゲットデバイス上の通信インタフェース（IrDA、USB およびシリアル）を通じてプリンタに対してコマンドやデータを容易に送信することができます。また、プリンタステータスを取得することができます。

本ライブラリでは、以下の機能を提供します。

- プリンタとの接続および切断
- プリンタへのデータ送信（印字データやコマンド*1）
- プリンタへのデータファイル送信（印字データやコマンド*1）
- プリンタステータスの取得
- プリンタのデータ待ち状態解除

(*1) プリンタから応答を取得するコマンドには対応しておりません。

1.2 対応する.NET Compact Framework

1.2.1 .NET Compact Framework のバージョンと対応製品

本ライブラリは対応する.NET Frameworkのバージョンにより.NET Compact Framework 1.0 用と.NET Compact Framework 2.0 用の製品が存在します。各製品と対応する.NET Compact Framework のバージョンの関係を 表 1-1 に示します。

表 1-1

製品名	対応.NET Compact Framework
SII DPU-S445 NETCF10	.NET Compact Framework 1.0
SII DPU-S445 NETCF20	.NET Compact Framework 2.0

SII DPU-S445 NETCF10 は、.NET Compact Framework 1.0 にのみ対応し、.NET Compact Framework 2.0 では利用できません。同様に、SII DPU-S445 NETCF20 は、.NET Compact Framework 2.0 にのみ対応し、.NET Compact Framework 1.0 では利用できません。（図 1-2）

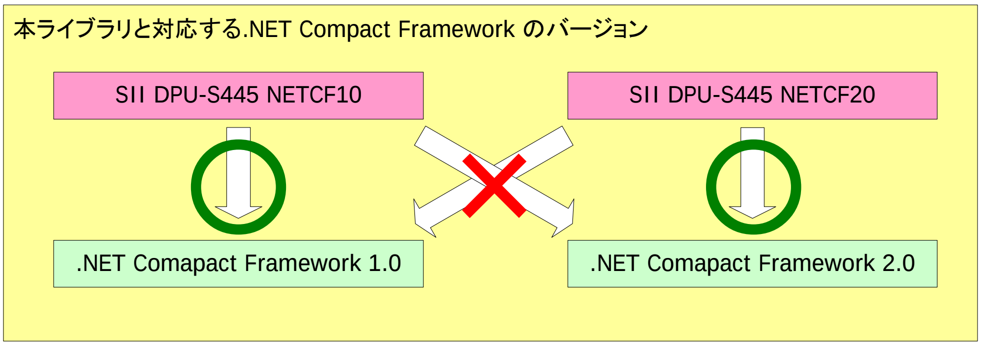


図 1-2

なお、本書では.NET Compact Framework の各バージョンに共通する項目の場合、本ライブラリを SII DPU-S445 NETCF と記述します。

2章　製品仕様

本章では、本ライブラリの製品仕様について説明します。

2.1 製品仕様

2.1.1 動作環境

本ライブラリの動作環境を 表 2-1 に示します。

表 2-1

対象 OS	Windows CE 5.0 Windows Embedded CE 6.0 （Windows Mobile 5.0、および、Windows Mobile 6 を含む）
対象 .NET Framework	.NET Compact Framework 1.0（SP3 以降） .NET Compact Framework 2.0（SP1 以降）
サポート言語	日本語 英語
対象通信方式	• 赤外線（IrDA DATA 1.2） • シリアル（RS-232C および Bluetooth（SPP）による仮想シリアルポート） • USB2.0
サポート開発言語	.NET Compact Framework を利用可能な開発言語 Visual C# .NET 2003（DPU-S445 NETCF10 のみ） Visual C# 2005（DPU-S445 NETCF10、DPU-S445 NETCF20） Visual Basic .NET 2003（DPU-S445 NETCF10 のみ） Visual Basic 2005（DPU-S445 NETCF10、DPU-S445 NETCF20）

2.1.2　通信方式の詳細

(1)　IrDA DATA 通信

通信速度は最大 115200 bps(SIR)です。アプリケーションプロトコルは IrLPT です。

(2)　シリアル通信

シリアル通信の仕様を 表 2-2 に示します。

表 2-2

ボーレート	9600、38400、57600、115200 bps
バイト長	8 ビット（固定）
ストップビット	1 ビット（固定）
パリティ	なし（固定）
フロー制御	Xon/Xoff、ハードウェア（CTS/RTS）

(3)　Bluetooth 通信

Bluetooth 通信は、SPP(Serial Port Profile)により仮想シリアルポートとして接続が確立されている場合に利用できます。

(4)　USB 通信

USB 通信を行う場合には、対象デバイスが USB ホスト機能をサポートしており、かつ対象 OS 上に USB Printer class driver（usbprn.dll）が実装されていることが必要です。

USB Printer class driver は、本ライブラリの対象 OS 上では LPT ポートとして扱えます。

USB Printer class driver は、本ライブラリには含まれておりません。USB Printer class driver の入手は、本ライブラリを利用するターゲットデバイスのメーカーから入手するか、Microsoft 社の Platform Builder にてターゲットデバイスに合った USB Printer class driver を作成してください。USB Printer class driver は、必ず本ライブラリを利用するターゲットデバイスの環境に合ったものをお使いください。

2.2 提供ファイルと使用方法

2.2.1 提供ファイル

本ライブラリは、プリントクラスライブラリ本体の DLL ファイルとシリアル通信用 DLL ファイルから構成されています。

対応する.NET Compact Framework のバージョンにより.NET Compact Framework 1.0 用と.NET Compact Framework 2.0 用の製品が存在します。

各バージョンの.NET Compact Frameworkに対応した製品とそのファイル構成を 表 2-3 に示します。

表 2-3

製品名	対応.NET Compact Framework	ファイル名	機能
SII DPU-S445 NETCF10	.NET Compact Framework 1.0	**SiiDpuS445NetCF10.dll**	クラスライブラリ本体
		SiiSerialNetCF10.dll	シリアル通信用
SII DPU-S445 NETCF20	.NET Compact Framework 2.0	**SiiDpuS445NetCF20.dll**	クラスライブラリ本体
		SiiSerialNetCF20.dll	シリアル通信用

2.2.2 本ライブラリのプロジェクトへの組み込み

本ライブラリを Visual C#、および、Visual Basic から利用するには、プロジェクトを作成し、本ライブラリ（**SiiDpuS445NetCF10.dll**、または、**SiiDpuS445NetCF20.dll**）を [参照の追加] で参照設定に追加します。（Visual C#.NET 2003、および、Visual Basic.NET 2003 では、 **SiiDpuS445NetCF10.dll** のみ利用可能です。）

2.2.3 作成したアプリケーションのターゲットデバイスでの実行方法

Visual C#、または、Visual Basic にて作成したアプリケーションと本ライブラリは同じフォルダに入れてください。

.NET Compact Framework 1.0 利用の場合は、作成したアプリケーションと同一フォルダに **SiiDpuS445NetCF10.dll** と **SiiSerialNetCF10.dll** を入れてください。

.NET Compact Framework 2.0 利用の場合は、作成したアプリケーションと同一フォルダに **SiiDpuS445NetCF20.dll** と **SiiSerialNetCF20.dll** を入れてください。

2.2.4 その他

開発するプログラムの種類によって、別途、下記の SDK 等が必要となる場合があります。

- Windows Mobile 5 Pocket PC SDK
- Windows Mobile 6 Standard SDK
- Windows Mobile 6 Professional SDK

これらの SDK 等は Microsoft 社のウェブサイトより入手してください。

3章　インストール

本章では、本ライブラリのインストールについて説明します。

3.1 インストーラ

3.1.1 インストーラについて

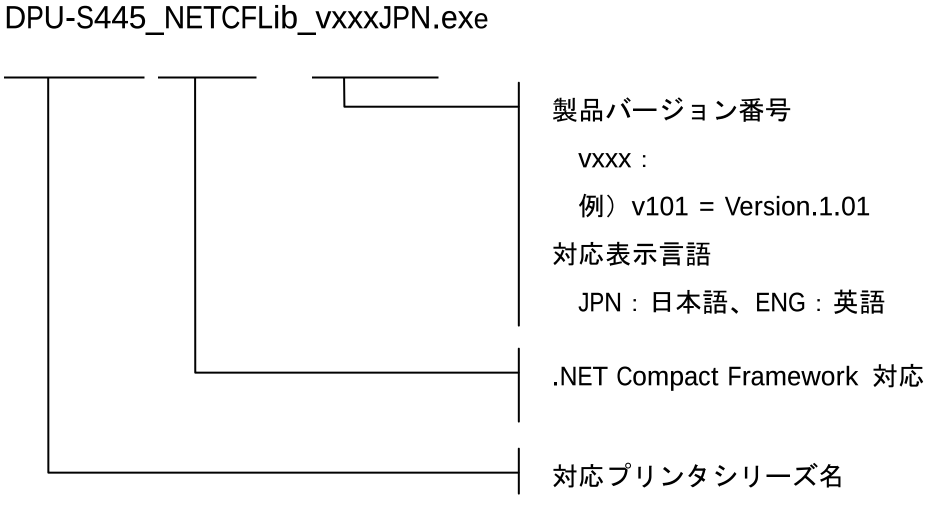
本クラスライブラリのインストーラには、日本語版インストーラと英語版インストーラが存在します（表 3-1）。日本語版と英語版の違いはインストール時に表示されるダイアログ画面の表示言語のみです。PC にインストールされるライブラリファイルに違いはありません。

表 3-1

対応言語	ファイル名
日本語	DPU-S445_NETCFLib_vxxxJPN.exe
英語	DPU-S445_NETCFLib_vxxxENG.exe

インストーラを実行すると、本ライブラリ（DPU-S445 NETCF10 と DPU-S445 NETCF20）が PC にインストールされます。

インストーラのファイル名は以下の内容を示します。



3.1.2　インストールファイルについて

インストーラを実行後、ＰＣにインストールされるファイルと、そのディレクトリ構成を 図 3-1 に示します。

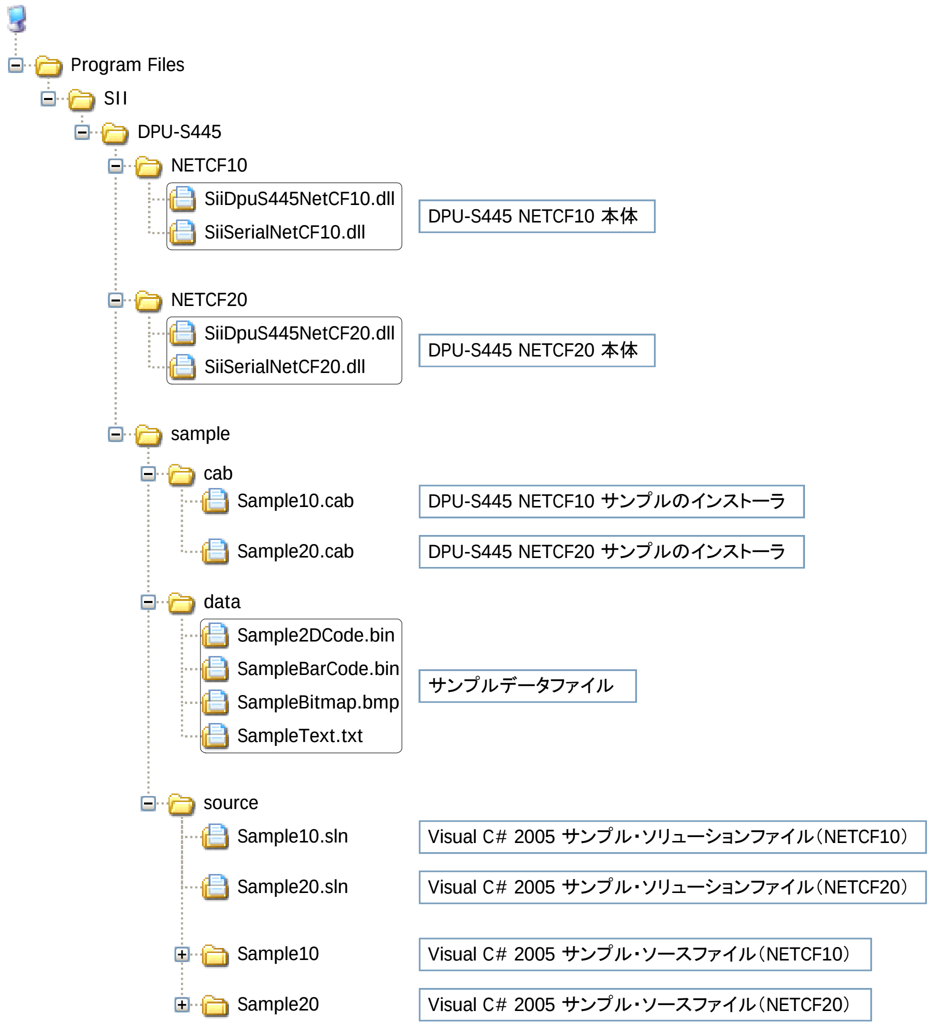


図 3-1

4章　プリントクラスライブラリの機能

本章では、本ライブラリのクラス構造、および、そのクラスに実装されているメソッドやプロパティに関して説明します。

本ライブラリの機能は、特に断りのない場合、DPU-S445 NETCF10、DPU-S445 NETCF20 共通です。

本ライブラリの具体的な利用方法に関しては、サンプルプログラム（C#）を用意していますのでそちらを参照してください。

4.1　クラスの概要

4.1.1　ネームスペースとクラス名

本ライブラリでは、対応する.NET Compact Frameworkに関わらずネームスペースやクラス名は共通です。表 4-1 にネームスペース、クラス名を示します。

表 4-1

ネームスペース	SII.SII_Print
クラス名	DPU_S445
完全修飾名	SII.SII_Printer.DPU_S445

4.1.2　クラスの構造

DPU_S445 クラス（以下、本クラス）の構造を図 4-1 に示します。

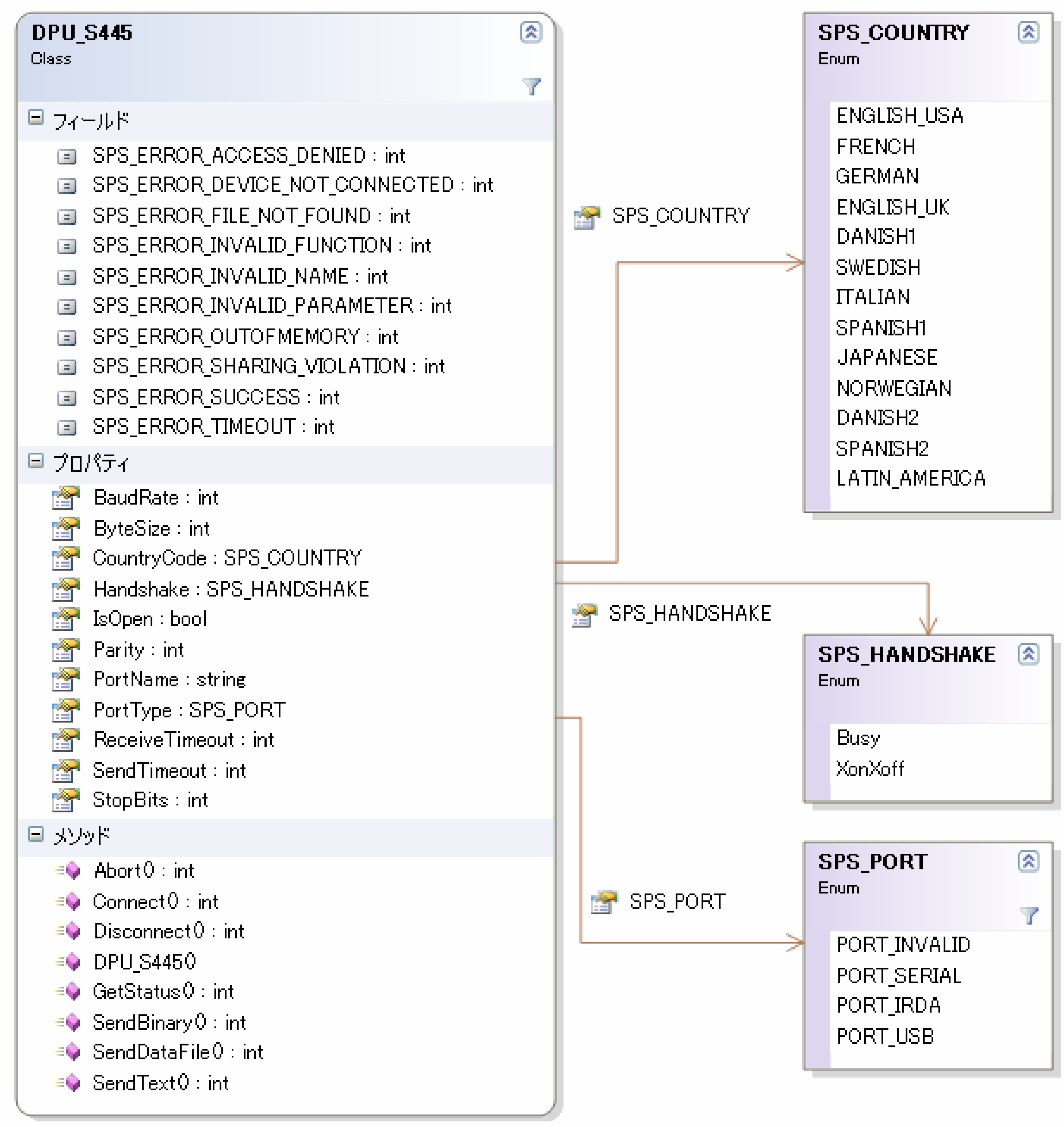


図 4-1

4.1.3　メソッド一覧

本クラスに実装されているメソッドの一覧を 表 4-2 に示します。

表 4-2

メソッド	機能概要	記載ページ
DPU_S445	コンストラクタ	4-4
Connect	プリンタとの接続	4-5
Disconnect	接続中のポートを切断	4-7
GetStatus	プリンタステータス取得	4-8
SendText	テキストデータ送信	4-10
SendBinary	バイナリデータ送信	4-11
SendDataFile	データファイル送信	4-12
Abort	プリンタのデータ待ち状態解除	4-15

4.1.4　プロパティ一覧

本クラスに実装されているプロパティの一覧を 表 4-3 に示します。

表 4-3

プロパティ	アクセス	機能概要	記載ページ
SendTimeout	R/W	データ送信時のタイムアウト時間	4-16
ReceiveTimeout	R/W	データ受信時のタイムアウト時間	4-17
CountryCode	R/W	国コード設定	4-18
BaudRate	R/W	シリアル通信時のボーレート	4-20
Handshake	R/W	シリアル通信時のフロー制御	4-21
ByteSize	R	シリアル通信時のバイト長	4-22
StopBits	R	シリアル通信時のストップビット	4-23
Parity	R	シリアル通信時のパリティ	4-24
PortName	R	接続中のポート名	4-25
PortType	R	接続中のポート種別	4-26
IsOpen	R	接続状態	4-27

4.2　メソッド詳細

本クラスに実装されている各メソッドの詳細について説明します。

4.2.1　コンストラクタ

DPU_S445()	コンストラクタ

本クラスの新たなオブジェクトを作成します。

構文

DPU_S445 ()

パラメータ

なし

戻り値

なし

解説

本クラスの新たなオブジェクトを作成します。

（例）Visual C#では以下のように宣言し、オブジェクトを作成します。

```
using SII.SII_Print;
    .
    .
DPU_S445 prn = new DPU_S445();
```

（例）Visual Basic では以下のように宣言し、オブジェクトを作成します。

```
Imports SII.SII_Print
    .
    .
Dim prn As New DPU_S445
```

4.2.2　Connectメソッド

Connect	プリンタとの接続

本クラスで利用する通信ポートを指定し、プリンタに接続します。

構文

Int32 ***Connect*** (String *PortName*)

パラメータ

PortName : プリンタと接続するポート名を指定します。String 型です。

*PortName*として有効なパラメータを 表 4-4 に示します。

表 4-4

PortName	ポート種別	説明
COMx:	シリアルポート	プリンタとシリアルポートによる接続を行います。 COM1:～COM9:の範囲で指定してください。 Bluetooth(SPP)による仮想シリアルポート接続の場合は、Bluetooth による接続が確立している COMx:を指定してください。
IR:	赤外線ポート (IrDA)	プリンタと IrDA による接続を行います。
LPTx:	プリンタポート （USB）	プリンタと USB ポートによる接続を行います。 LPT1:～LPT9:の範囲で指定してください。

戻り値

本メソッドの戻り値を 表 4-5 に示します。

表 4-5

定数	値	説明
SPS_ERROR_SUCCESS	0	正常に実行されました。
SPS_ERROR_ACCESS_DENIED	5	ハンドルの取得に失敗しました。 使用できないポートを指定しました。
SPS_ERROR_SHARENG_VIOLATION	32	既にオープン済みのポートを指定しました。 指定されたポートは他のデバイスがオープン中です。
SPS_ERROR_INVALID_NAME	123	指定されたポートが存在しません。
SPS_ERROR_DEVICE_NOT_CONNECTED	1167	プリンタが見つかりません。（IrDA の場合）
SPS_ERROR_TIMEOUT	1467	送信、または受信タイムアウトが発生しました。

解説

本メソッドは、他の本クラスのメソッドを使用する前に呼び出します。

本メソッドは、*PortName* で指定されたポートをオープンし、プリンタへの接続を行います。USB ポート以外で接続を行う場合は、事前にプリンタの機能設定にて通信モードを *PortName* で指定するポート種別と一致させてください。シリアルポートで接続を行う場合には、本メソッド呼び出し前に ***Baudrate*** プロパティおよび ***Handshake*** プロパティを、プリンタの通信設定に合わせて設定してください。

本ライブラリを正しく動作させるため、本メソッドは接続時にプリンタの設定を変更する場合があります。

参照

Baudrateプロパティ、Handshakeプロパティ

4.2.3 Disconnectメソッド

Disconnect 接続中のポートを切断

接続中のポートを切断します。

構文

Int32 ***Disconnect*** ()

パラメータ

なし

戻り値

本メソッドの戻り値は、表 4-6 の通りです。

表 4-6

定数	値	説明
SPS_ERROR_SUCCESS	0	正常に実行されました。
SPS_ERROR_INVALID_FUNCTION	1	ポートがオープンされておりません。
SPS_ERROR_ACCESS_DENIED	5	クローズするハンドルに誤りがありました。

解説

本メソッドは、接続中のポートを切断します。ポートの切断後もプロパティの設定は可能です。

4.2.4 GetStatusメソッド

GetStatus	プリンタステータス取得

プリンタステータスを取得します。

構文

Int32 ***GetStatus*** (ref Int32 *Status*)

パラメータ

Status : プリンタステータスを保存するバッファを指定します。バッファの型は Int32 型です。

有効なステータスは下位 8 ビットです。

プリンタステータスの各ビットが示す内容は、表 4-7 と 表 4-8 の通りです。

なお、ステータスの取得に失敗した場合は、*Status* には 0 が格納されます。

表 4-7

ビット	機能	値	
		0	**1**
0	紙無しエラー	OK	エラー
1	ヘッドアップ	OK	エラー
2	VP 電圧異常	OK	エラー
3	ヘッド温度異常	OK	エラー
4	機能設定エラー	OK	エラー
5	バッテリ電圧の状態	表 4-8	
6	バッテリ電圧の状態		
7	予約	-	固定

表 4-8

ビット 6	ビット 5	バッテリの状態
0	0	8.0V 以上
0	1	7.5V 以上 8.0V 未満
1	0	7.0V 以上 7.5V 未満
1	1	7.0V 未満

戻り値

本メソッドの戻り値は、表 4-9 の通りです。

表 4-9

定数	値	説明
SPS_ERROR_SUCCESS	0	正常に実行されました。
SPS_ERROR_INVALID_FUNCTION	1	ポートがオープンされていません。
SPS_ERROR_DEVICE_NOT_CONNECTED	1167	ステータスの取得に失敗しました。（IrDA、LPT） （プリンタが接続されていません。）

解説

プリンタステータスを取得します。取得したプリンタステータスは *Status* に格納されます。格納された *Status* のうち有効なデータは下位 8 ビットです。また、ステータスの取得に失敗した場合は、*Status* に 0 が格納されます。

本ライブラリでは、接続が確立している間、プリンタから送信されるステータスデータをライブラリ内部のバッファに自動的に保持します。プリンタは、プリンタの状態が変化する度にステータスデータを送信しますが、本ライブラリはその度にバッファを自動的に更新し、常に最新のステータスデータを保持します。本メソッドでは、その保持しているステータスデータを *Status* に格納します。

ただし、バッテリの状態（ビット 5、ビット 6）は、他のステータスビットが変化した場合のみ更新されます。

4.2.5 SendTextメソッド

SendText	テキストデータ送信

指定されたテキストデータを ***CountryCode*** プロパティにしたがってエンコードし、プリンタに送信します。

構文

Int32 ***SendText*** (String *s*)

パラメータ

s　テキストデータを指定します。String 型です。

指定可能なデータサイズは最大 3840 バイトです。

戻り値

本メソッドの戻り値は、表 4-10 の通りです。

表 4-10

定数	値	説明
SPS_ERROR_SUCCESS	0	正常に実行されました。
SPS_ERROR_INVALID_FUNCTION	1	ポートがオープンされていません。
SPS_ERROR_INVALID_PARAMETER	87	データサイズが 0 バイトのデータが指定されました。
		データサイズが 3840 バイトを超えました。
		テキストデータのエンコードでエラーが発生しました。
SPS_ERROR_DEVICE_NOT_CONNECTED	1167	プリンタが接続されていません。(IrDA)
SPS_ERROR_TIMEOUT	1460	送信、または受信タイムアウトが発生しました。

解説

パラメータ *s* で指定されたテキストデータをプリンタに送信します。指定可能なデータサイズは最大 3840 バイトです。

指定されたテキストデータは ***CountryCode*** プロパティにしたがってエンコードされ、プリンタに送信されます。エンコードに関する詳細は ***CountryCode*** プロパティを参照してください。

本メソッドでは、テキストデータの最後に改行コードは付加しません。したがって指定データの最後に改行コード (CR+LF) が付加されていない場合、プリンタの行バッファにデータが残り、送信したテキストデータの一部、または、全部が印字されない場合があります。

参照

CountryCodeプロパティ

4.2.6　SendBinaryメソッド

SendBinary	バイナリデータ送信

構文

Int32 ***SendBinary*** (Byte[] *b*)

パラメータ

b　バイナリデータを指定します。Byte[]型です。

指定可能なデータサイズは最大 3840 バイトです。

戻り値

本メソッドの戻り値は、表 4-11 の通りです。

表 4-11

定数	値	説明
SPS_ERROR_SUCCESS	0	正常に実行されました。
SPS_ERROR_INVALID_FUNCTION	1	ポートがオープンされていません。
SPS_ERROR_INVALID_PARAMETER	87	データサイズが 0 バイトのデータが指定されました。
		データサイズが 3840 バイトを超えました。
SPS_ERROR_DEVICE_NOT_CONNECTED	1167	プリンタが接続されていません。(IrDA)
SPS_ERROR_TIMEOUT	1460	送信、または受信タイムアウトが発生しました。

解説

パラメータ *b*で指定されたバイナリデータをプリンタに送信します。指定可能なデータサイズは最大 3840 バイトです。

本メソッドは、指定されたデータを無変換のままプリンタに送信します。

本メソッドでプリンタコマンドをバイナリデータとして送信することにより、本ライブラリではサポートされていないプリンタ機能を利用することができます。ただし、プリンタから応答を取得するコマンドには対応しておりません。

4.2.7 **SendDataFileメソッド**

SendDataFile	データファイル送信

指定したテキストファイル、ビットマップファイル、バイナリデータファイルを送信します。

構文

Int32 ***SendDataFile*** (String *fn*)

パラメータ

fn データファイル名を指定します。String 型です。

指定可能なファイルサイズは最大 1048576 バイト (1M バイト)です。

指定可能なデータファイルの拡張子は 表 4-12 の通りです。

表 4-12

拡張子	ファイル形式	説明
.txt	テキストファイル	指定したファイル中のテキストデータを ***CountryCode*** プロパティにしたがってエンコードし、プリンタに送信します。
.bmp	ビットマップファイル	指定したファイルのビットマップデータをラスタビットイメージに変換し、プリンタに送信します。指定可能なビットマップデータはモノクロ（2 値）のみサポートされています。
.bin .dat	バイナリファイル	指定したファイル中のバイナリデータを無変換のままプリンタに送信します。

戻り値

本メソッドの戻り値は、表 4-13 の通りです。

表 4-13

定数	値	説明
SPS_ERROR_SUCCESS	0	正常に実行されました。
SPS_ERROR_INVALID_FUNCTION	1	ポートがオープンされていません。
SPS_ERROR_FILE_NOT_FOUND	2	指定されたファイルが存在しません。
SPS_ERROR_OUTOFMEMORY	14	ビットマップデータの読み込みでメモリ不足が発生しました。（ビットマップファイル指定時）
SPS_ERROR_INVALID_PARAMETER	87	データサイズが0バイトのファイルが指定されました。
		データサイズが 1048576 バイトを超えるファイルが指定されました。
		1 行のデータサイズが 3840 バイトを超えました。（テキストファイル指定時）
		テキストデータのエンコード時にエラーが発生しました。（テキストファイル指定時）
		未サポートのビットマップ形式データが指定されました。（ビットマップファイル指定時）
		紙幅方向が 65535 ドットを超えるデータが指定されました。（ビットマップファイル指定時）
		紙送り方向が 4000 ドットを超えるデータが指定されました。（ビットマップファイル指定時）
		紙幅方向と紙送り方向の双方が 832 ドットを超えるデータが指定されました。（ビットマップファイル指定時）
SPS_ERROR_DEVICE_NOT_CONNECTED	1167	プリンタが接続されていません。(IrDA)
SPS_ERROR_TIMEOUT	1460	送信、または受信タイムアウトが発生しました。

解説

パラメータ *fn* で指定されたファイルは、ファイル名の拡張子によりファイル形式が判定されます。本メソッドで有効なファイル形式は、テキストファイル、ビットマップファイル、バイナリファイルです。本メソッドではファイル形式に応じて以下の処理を行います。

① テキストファイルの処理

テキストファイル中のデータは 1 行単位に ***CountryCode*** プロパティにしたがってエンコードされ、プリンタに送信されます。1 行に指定可能なデータサイズは最大 3840 バイトです。エンコードに関する詳細は ***CountryCode*** プロパティを参照してください。

本メソッドでは、テキストデータの最後に改行コードは付加しません。したがって指定ファイルの最後に改行コード (CR+LF) が付加されていない場合、プリンタの行バッファにデータが残り、送信したテキストデータの一部、または、全部が印字されない場合があります。

② ビットマップファイルの処理

ビットマップファイルはモノクロ（2 値）のビットマップデータのみサポートします。なお、本メソッドでは、紙幅方向および紙送り方向の双方が 832 ドットを超えるビットマップデータは印字できません。また、データサイズが 1048576 バイト（1M バイト）を超えるビットマップデータも印字できません。

本メソッドを実行する場合は、ご利用環境に十分なメモリが実装されていることが必要です。

なお、.NET Compact Framework 1.0 環境＋DPU-S445 NETCF10 ではイメージ展開処理に時間がかかるため、ビットマップの印刷に時間がかかる場合があります。ビットマップデータの印刷を主として行う場合には、.NET Compact Framework 2.0 環境＋DPU-S445 NETCF20 での利用を推奨します。

ビットマップデータの送信が何らかの理由で中断した場合、プリンタは指定したビットマップデータのサイズまで他のメソッドをビットマップデータの続きと誤認識する場合があります。このような状態が発生した場合は、***Abort*** メソッドにてプリンタのビットマップデータ待ち状態を解除してください。

③ バイナリファイルの処理

バイナリファイルは無変換のままプリンタに送信されます。本メソッドでプリンタコマンドのバイナリデータファイル送信することにより、本ライブラリではサポートされていないプリンタ機能を利用することができます。ただし、プリンタから応答を取得するコマンドには対応しておりません。

参照

CountryCodeプロパティ、Abortメソッド

4.2.8　Abortメソッド

Abort	プリンタのデータ待ち状態解除

プリンタがビットマップデータ待ち状態となっているのを解除します。

構文

Int32 ***Abort*** ()

パラメータ

なし

戻り値

本メソッドの戻り値は、表 4-14 の通りです。

表 4-14

定数	値	説明
SPS_ERROR_SUCCESS	0	正常に実行されました。
SPS_ERROR_INVALID_FUNCTION	1	ポートがオープンされていません。
SPS_ERROR_TIMEOUT	1460	送信、または受信タイムアウトが発生しました。

解説

SendDataFile メソッドによるビットマップデータの送信が中断した場合、プリンタは指定したビットマップデータの受信完了まで、他の処理を受け付けません。（メソッドや送信データが誤解釈され、ビットマップデータの続きと認識されます。）この状況を解消するために、本メソッドを利用し、プリンタのデータ待ち状態を解除します。なお、本メソッドを実行した場合、未印字のビットマップデータの一部が印字される場合があります。

参照

SendDataFileメソッド（ビットマップファイル）

4.3 プロパティ詳細

各プロパティの詳細について説明します。

4.3.1 SendTimeoutプロパティ

SendTimeout	送信タイムアウト時間

データ送信時のタイムアウト時間を msec(ミリ秒)単位で設定します。

構文

SendTimeout

アクセス	読み書き両用
データ型	System.Int32
初期値	10000 （msec） （10 秒）
範囲	1 ～ 60000 （msec）（60 秒）

解説

本プロパティはプリンタとの接続中/切断中を問わず設定可能です。ただし、設定内容は即座には反映されません。本プロパティ設定後の ***Connect*** メソッド実行時に設定が反映されます。そのため、プリンタと接続中に設定を変更する場合は、一旦、***Disconnect*** メソッドで切断後、再度 ***Connect*** メソッドを実行してください。

4.3.2 **ReceiveTimeout**プロパティ

ReceiveTimeout	受信タイムアウト時間

データ受信時のタイムアウト時間を msec(ミリ秒)単位で設定します。

構文

ReceiveTimeout

アクセス　　読み書き両用

データ型　　System.Int32

初期値　　10000（msec）（10 秒）

範囲　　1 ～ 60000 （msec）（60 秒）

解説

本プロパティはプリンタとの接続中/切断中を問わず設定可能です。ただし、設定内容は即座には反映されません。本プロパティ設定後の ***Connect*** メソッド実行時に設定が反映されます。そのため、プリンタと接続中に設定を変更する場合は、一旦、***Disconnect*** メソッドで切断後、再度 ***Connect*** メソッドを実行してください。

4.3.3 CountryCodeプロパティ

CountryCode	国コード

プリンタの国際文字セットや文字セットの設定、テキストデータのエンコードパラメータのキーとなる国コードを設定/取得します。

構文

CountryCode

アクセス　読み書き両用

データ型　SII.SII_Printer.SPS_COUNTRY

初期値　JAPANESE　.NET Compact Framework が日本語をサポートしている場合

ENGLISH_USA　.NET Compact Framework が日本語を未サポートの場合

範囲

本プロパティで設定可能な国コード定数は、表 4-15 の通りです。

表 4-15

定数 （SPS_COUNTRY）	値	国際文字セット （プリンタ）	文字セット （プリンタ）	エンコード
ENGLISH_USA	0	アメリカ	拡張グラフィックス文字セット	IBM437 *1
FRENCH	1	フランス	拡張グラフィックス文字セット	IBM437 *1
GERMAN	2	ドイツ	拡張グラフィックス文字セット	IBM437 *1
ENGLISH_UK	3	イギリス	拡張グラフィックス文字セット	IBM437 *1
DANISH1	4	デンマークⅠ	拡張グラフィックス文字セット	IBM437 *1
SWEDISH	5	スウェーデン	拡張グラフィックス文字セット	IBM437 *1
ITALIAN	6	イタリア	拡張グラフィックス文字セット	IBM437 *1
SPANISH1	7	スペイン	拡張グラフィックス文字セット	IBM437 *1
JAPANESE	8	日本	カタカナ文字セット	shift_jis *2
NORWEGIAN	9	ノルウェー	拡張グラフィックス文字セット	IBM437 *1
DANISH2	10	デンマークⅡ	拡張グラフィックス文字セット	IBM437 *1
SPANISH2	11	スペインⅡ	拡張グラフィックス文字セット	IBM437 *1
LATIN_AMERICA	12	ラテンアメリカ	拡張グラフィックス文字セット	IBM437 *1

解説

本プロパティは常時設定可能です。***SendText***メソッドや***SendDataFile***メソッドで指定したテキストデータは、本プロパティの設定によりプリンタの国際文字セットと文字セットを設定し、テキストデータは 表 4-15 のエンコード形式で変換されプリンタに送信されます。国際文字セット、および、文字セットについては、プリンタ本体の技術説明書を参照してください。

ご利用環境に導入されている.NET Compact Framework が日本語に対応していない場合は、ENGLISH_USA の扱いとなります。

(*1)　国コードが JAPANESE 以外の場合、エンコードの際に、プリンタ側の拡張グラフィックス文字セットに該当する文字が無い場合は、'?' (3Fh)が印字されます。

(*2)　国コードが JAPANESE の場合、JIS 第 1、第 2 水準の漢字が印字できます。
ただし、シフト JIS 8040h～879Ch の文字はプリンタに割り当てられた文字で印字されます。
また、シフト JIS ED40h～EDFFh、EE40h～EEFFh、FA40h～FAFFh、FB40h～FBFFh、FC40h～FC4Fh の文字は全角スペース（シフト JIS 8140h）で印字されます。

参照

SendText**メソッド*、SendDataFile**メソッド*

4.3.4 Baudrateプロパティ

Baudrate	通信速度（ボーレート）

シリアル通信時のボーレートを設定/取得します。

構文

Baudrate

アクセス　　読み書き両用

データ型　　System.Int32

初期値　　115200 (bps)

範囲

本プロパティで設定可能なボーレートは、表 4-16 の通りです。

表 4-16

値	説明
9600	9600 bps
19200	19200 bps
38400	38400 bps
57600	57600 bps
115200	115200 bps

解説

シリアル通信時のボーレートを設定/取得します。***Connect*** メソッドは、本プロパティの設定値に合わせてシリアル通信時のボーレートを設定します。なお、プリンタ本体では上記以外のボーレートの設定が可能ですが、本ライブラリでは上記以外のボーレートはサポートいたしません。

本プロパティはプリンタとの接続中/切断中を問わず設定可能です。ただし、設定内容は即座には反映されません。本プロパティ設定後の ***Connect*** メソッド実行時に設定が反映されます。そのため、プリンタと接続中に設定を変更する場合は、一旦、***Disconnect*** メソッドで切断後、再度 ***Connect*** メソッドを実行してください。

4.3.5 Handshakeプロパティ

Handshake	フロー制御

シリアル通信時のフロー制御方法を設定/取得します。

構文

Handshake

アクセス　　読み書き両用

データ型　　SII.SII_Printer.SPS_HANDSHAKE

初期値　　Busy

範囲

本プロパティで設定可能なフロー制御の定数は、表 4-17 の通りです。

表 4-17

定数	値	説明
XonXoff	1	Xon/Xoff
Busy	2	ハードウェア

解説

シリアル通信時のフロー制御方法を設定/取得します。***Connect*** メソッドは、本プロパティの設定値に合わせてシリアル通信時のフロー制御方法を設定します。

本プロパティはプリンタとの接続中/切断中を問わず設定可能です。ただし、設定内容は即座には反映されません。本プロパティ設定後の ***Connect*** メソッド実行時に設定が反映されます。そのため、プリンタと接続中に設定を変更する場合は、一旦、***Disconnect*** メソッドで切断後、再度 ***Connect*** メソッドを実行してください。

4.3.6　**ByteSize**プロパティ

ByteSize	バイト長

シリアル通信時のバイト長を取得します。

構文

ByteSize

アクセス　　読み取り専用

データ型　　System.Int32

初期値　　8　（固定）

解説

本プロパティはシリアル通信時のバイト長を取得します。

本プロパティは読み取り専用のため値を設定することはできません。

4.3.7　StopBitsプロパティ

StopBits	ストップビット長

シリアル通信時のストップビット長を取得します。

構文

StopBits

アクセス　　読み取り専用

データ型　　System.Int32

初期値　　1 （固定）

解説

本プロパティはシリアル通信時のストップビット長を取得します。

本プロパティは読み取り専用のため値を設定することはできません。

4.3.8　Parityプロパティ

Parity	パリティ

シリアル通信時のパリティを取得します。

構文

Parity

アクセス　　読み取り専用

データ型　　System.Int32

初期値　　0 （固定）（パリティなし）

解説

本プロパティはシリアル通信時のパリティを取得します。

本プロパティは読み取り専用のため値を設定することはできません。

4.3.9　**PortNameプロパティ**

PortName	接続中のポート名

接続中のポート名を取得します。

構文

PortName

アクセス　　読み取り専用

データ型　　System.String

初期値　　“”（空文字列）

解説

本プロパティは ***Connect*** メソッドで指定された接続中のポート名を取得します。

本プロパティは読み取り専用のため値を設定することはできません。

4.3.10 **PortType**プロパティ

PortType	接続中のポート種別

接続中のポート種別を取得します。

構文

PortType

アクセス　　読み取り専用

データ型　　SII.SII_Printer.SPS_PORT

初期値　　PORT_INVALID

範囲

本プロパティで取得可能なポート種別の定数は、表 4-18 の通りです。

表 4-18

定数(SPS_PORT)	値	説明
PORT_INVALID	0	接続されていません。
PORT_SERIAL	1	COMx:で接続されています。
PORT_IRDA	2	IrDA で接続されています。
PORT_USB	3	LPTx:で接続されています。

解説

本プロパティは、接続中のポート種別を取得します。

本プロパティは読み取り専用のため値を設定することはできません。

4.3.11 IsOpenプロパティ

IsOpen	接続状態

本ライブラリの接続状態を取得します。

構文

IsOpen

アクセス　　読み取り専用

データ型　　System.Boolean

初期値　　False

範囲

本プロパティで取得可能な接続状態は、表 4-19 の通りです。

表 4-19

定数	説明
True	接続状態（ポートはオープン状態）
False	非接続状態（ポートはクローズ状態）

解説

本プロパティは、本ライブラリでのプリンタとの接続の有無を取得します。

本プロパティは読み取り専用のため値を設定することはできません。

5章　サンプルプログラム

本章では、本ライブラリで提供するサンプルプログラムとそのソースファイルについて説明します。

5.1　提供サンプル

5.1.1　サンプルプログラム

本ライブラリでは、Microsoft Windows Mobile 5.0 デバイス上で動作する本ライブラリを利用したサンプルプログラムを提供しています。

サンプルプログラムは、本ライブラリをインストール後、PC の `¥Program Files¥SII¥DPU-S445¥sample¥cab` フォルダ以下に Sample10.cab および Sample20.cab としてインストールされています。（図 3-1 参照）

サンプルプログラムのcabファイルが対応する.NET Compact Frameworkのバージョンを 表 5-1 に示します。

表 5-1

cab ファイル	対応.NET Compact Framework
Sample10.cab	.NET Compact Framework 1.0
Sample20.cab	.NET Compact Framework 2.0

Windows Mobile 5.0 デバイスは、標準では.NET Compact Framework 2.0 がインストールされていません。Sample20.cab を利用する場合は、別途、Microsoft 社のウェブサイトよりダウンロードして事前にインストールしてください。

5.1.2　サンプルプログラムのインストール

サンプルプログラムを Microsoft Windows Mobile デバイスにインストールするには、cab ファイル（Sample10.cab または Sample20.cab）を Windows Mobile デバイス上の適当なフォルダにコピーし、ファイルエクスプローラから cab ファイルを実行してください。

図 5-1 に Sample10.cab を実行し、Windows Mobile デバイスの [プログラム] に SII Sample10 のアイコンが登録された例を示します。

図 5-1

5.1.3　サンプルプログラムの実行

サンプルプログラムを実行すると、図 5-2 の画面が表示されます。(Sample20 の画面も同様となります。)

各機能を 表 5-2 に示します。

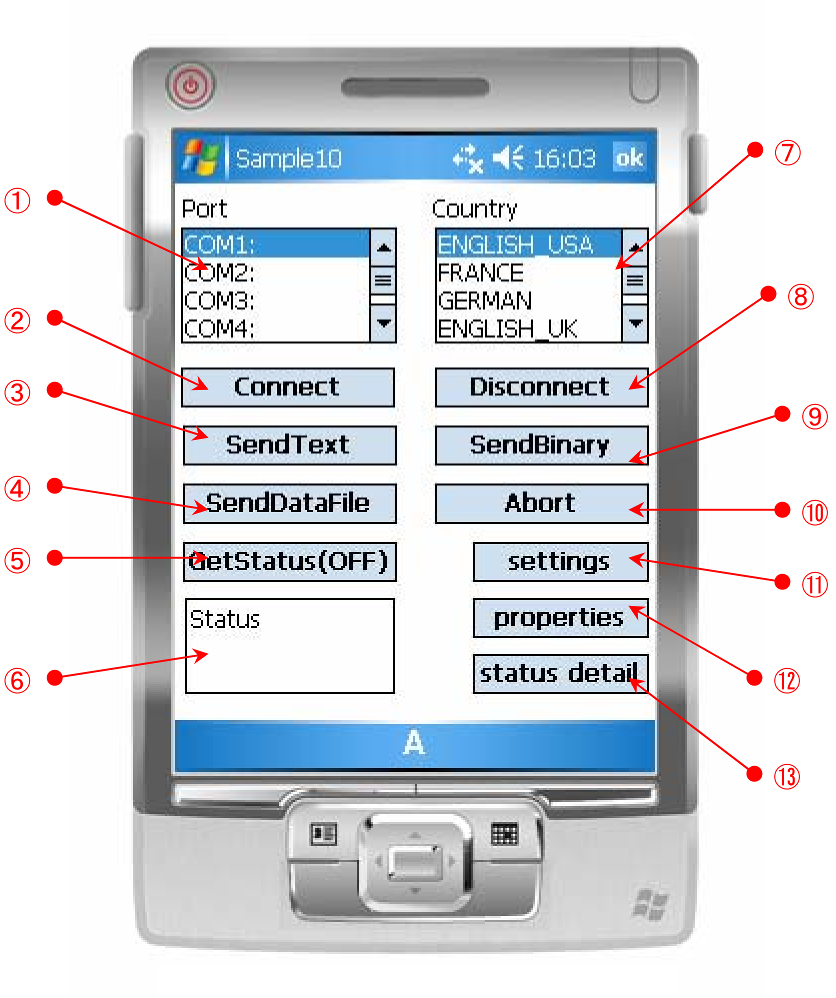


図 5-2

表 5-2

番号	説明	備考
①	ポート選択	***Connect*** メソッドのパラメータを COM1:～COM9:、LPT1:、LPT2:、IR:から選択します。
②	***Connect*** メソッド	①で選択されたポートをパラメータにして ***Connect*** メソッドを実行します。
③	***SendText*** メソッド	***SendText*** メソッドを実行します。別ウィンドウでテキスト入力画面が表示されます。
④	***SendDataFile*** メソッド	***SendDataFile*** メソッドを実行します。別ウィンドウでファイル選択画面が表示されます。
⑤	***GetStatus*** メソッド	タイマーによる ***GetStatus*** メソッドの定期的実施、または停止を指定します。 （ON の場合は定期的実施、OFF の場合は停止となります。） 取得したステータスは⑥に表示されます。 定期実施時のタイマー間隔は 500msec です。
⑥	***GetStatus*** メソッドの結果	⑤が ***GetStatus***（ON）の場合のステータスを表示します。
⑦	***CountryCode*** プロパティ	***CountryCode*** プロパティを選択します。
⑧	***Disconnect*** メソッド	***Disconnect*** メソッドを実行します。
⑨	***SendBinary*** メソッド	***SendBinary*** メソッドを実行します。別ウィンドウでバイナリデータ入力画面が表示されます。
⑩	***Abort*** メソッド	***Abort*** メソッドを実行します。
⑪	Settings	プロパティ設定画面を表示します。
⑫	Properties	プロパティの一覧画面を表示します。
⑬	Status Detail	***GetStatus*** メソッドで取得したプリンタステータス値の詳細を Status 画面に表示します。

5.1.4　サンプルプログラムのソースファイル

本ライブラリでは、サンプルプログラム Sample10 および Sample20 のソースファイル（Visual C# 2005）を提供しています。

ソースファイルは、本ライブラリをインストール後、PC の ¥Program Files¥SII¥DPU-S445¥sample ¥source フォルダ以下にインストールされています。（図 3-1 参照）

5.1.5　注意事項

サンプルプログラムに対しての動作保証およびサポートは行っておりません。

サンプルプログラムは予告なしに変更される場合があります。

6章　免責

本ソフトウェアは不具合の発生のなきよう十分留意して開発しておりますが、万一、本ソフトウェアによってお客様に損害を及ぼした場合、一切の責を負いかねますのでご了承ください。